Driveroo

Submitted by: Driverooers

Submitted to: Hao Lac
ICET Department,
School of Engineering Technology and Applied Science
Progress Campus, Block A
Centennial College

Discipline: Software Engineering / Game - Programming

Due Date: 2023-05-23

Declaration of Sole Authorship

We, Driverooers, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed:

- Muhammad Sabeeh, 301184564 (Game - Programming)
- Jin Huang, 301020707 (Game - Programming)
- Tannus Esquerdo, 301288342 (Software Eng Technology)

Date: 2023-05-23

Abstract

The proposed project presents Driveroo, a mobile application that seeks to revolutionize the process of scheduling driving lessons by giving consumers a practical and tailored experience. Users of the app may search for driving instructors nearby, look at teacher profiles and reviews, set up appointments, and conveniently book lessons using the app. It responds to the rising desire for a quicker, easier way to locate driving instructors, particularly in regions with high demand.

The importance of Driveroo lies in its capacity to simplify the booking procedure, increase accessibility for students, and broaden the visibility of driving instructors. The software streamlines the scheduling and payment procedures, making it simpler for users to book courses. It also integrates payment channels. Additionally, teacher discussions guarantee a unique learning path for every user.

Potential users should download the Driveroo app and use it to look for driving instructors, make appointments for consultations, and reserve lessons in order to take full advantage of its features. Users may locate and book driving instructors in a more effective and easy manner by making use of the app's features and functions. The number of registered driving instructors, customer contentment, and favourable app store reviews will be used to gauge the project's success since they show the app's influence and worth within the driving instruction sector.

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The technological issue addressed in this project is the slowness and absence of a simplified approach to booking driving lessons. The current techniques for locating and hiring driving instructors sometimes include time-consuming searches, restricted access to teacher information, and manual scheduling. The goal of this project is to create a mobile application called Driveroo to address this issue and provide a more effective way for learners to identify and book driving teachers.

This Technical Report (TR) describes the work done to propose and detail the development of the Driveroo app. The objective is to provide a user-friendly platform that streamlines the process of identifying and hiring driving instructors, enhancing the entire learning experience for students.

The report includes sections such as Introduction, Methodology and Results, Conclusions, Recommendations, and Appendices. It covers a literature review, a proposed solution, and user role modeling. The report focuses on planning, analysis, and user-centric aspects of the project, omitting specific implementation details. Procedures used include brainstorming, group work, user role modeling, release and iteration planning, progress monitoring, and acceptance testing. The scope encompasses the development of the Driveroo app and the processes involved in user role modeling, release planning, and iteration planning. It provides a comprehensive understanding of project objectives, user requirements, and the planning and testing processes involved in app development.

The primary goal is to create and introduce the Driveroo mobile application, which will let users look up driving instructors, review their profiles and reviews, schedule consultations, and book lessons all through the app. The emphasis on personalized experiences through talks with teachers prior to personalized scheduling of classes is what makes this concept distinctive.

2.0 METHODOLOGY AND RESULTS

2.1 Literature Review

Existing methods for identifying and booking driving instructors have shortcomings that Driveroo can overcome. These solutions are often tied to specific firms or driving schools, limiting independent teachers' access to new students. As a result, individual instructors have few possibilities to extend their student base beyond word-of-mouth recommendations and self-promotion. Furthermore, the existing process is inefficient and inconvenient. Users are frequently faced with a time-consuming process of picking appropriate packages, submitting required information, and being given random instructors without the chance to engage or discuss with them beforehand. Furthermore, scheduling is rigid and inflexible.

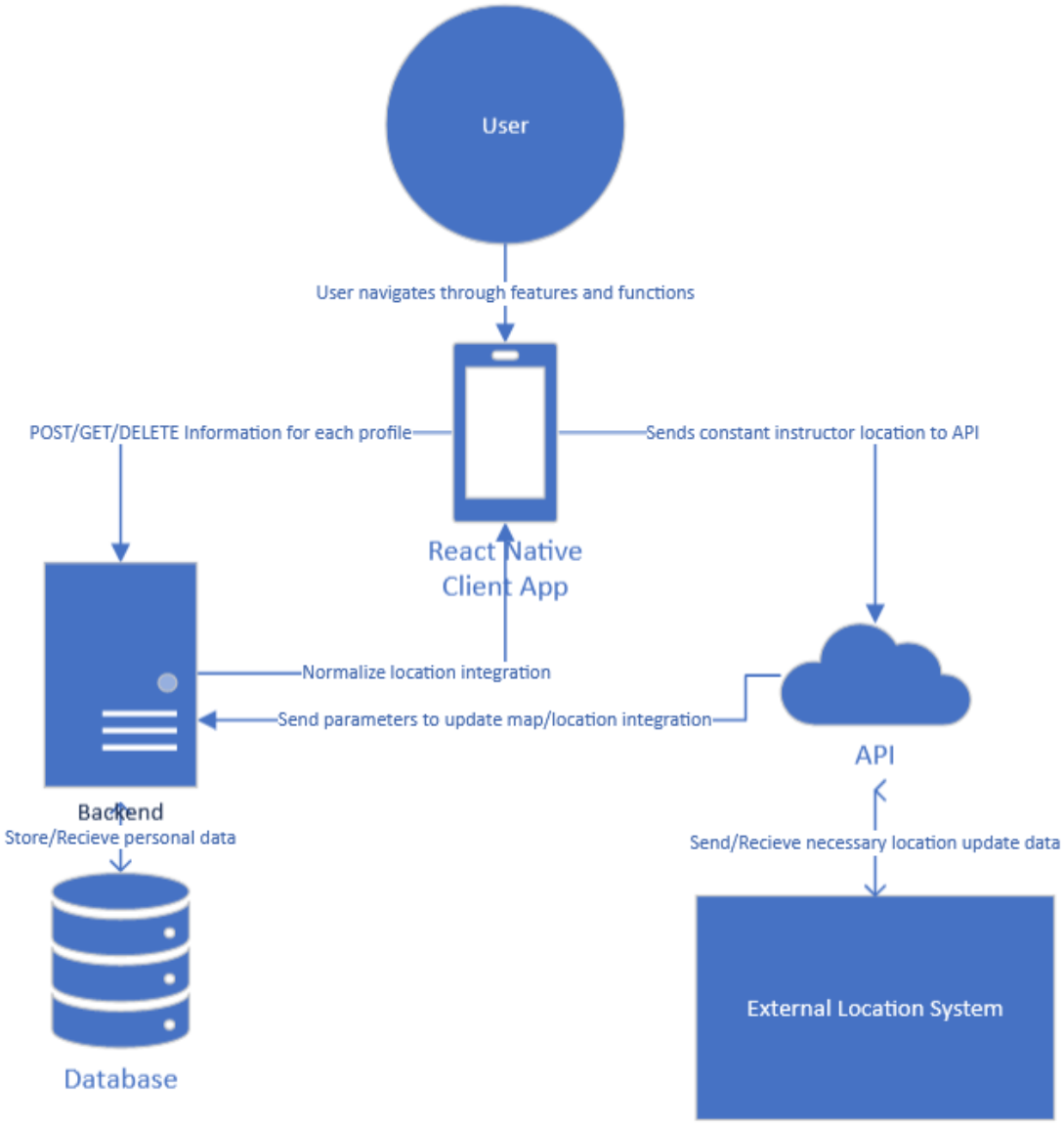
Despite these flaws, existing methods have some advantages, such as verification and recruitment processes. Driving school instructors are thoroughly vetted to ensure they have the required qualifications and credentials. Schools frequently hire instructors straight from graduating classes of teaching programs, resulting in a greater pool of trained teachers. These verification techniques instill

trust and professionalism in users by providing them with confidence in the teachers' abilities.

2.2 Proposed Solution

Our system's strength to greatly ease the process of finding driving instructors for both consumers and employees is one of its main advantages. Users may easily access a wide network of driving instructors in their area by using our application, which is based on their general location. Users benefit from increased convenience and time and effort savings because of this functionality. Additionally, our platform provides a great level of flexibility, enabling customers to communicate openly with teachers about session scheduling and specific needs. The platform includes a strong profile matching system as well, allowing for easy pairing based on things like particular needs, instructor ratings, financial concerns, and schedule compatibility. This capability helps create a highly effective and customised experience, which ultimately optimises the user journey.

There are a few flaws in our system that need to be considered. First, it depends on the user's area having a sufficient number of driving instructors. If there aren't enough instructors, consumers may not be able to book courses through the app**.** Furthermore, the lengthy verification process for driving instructors that is required by regulatory rules. Although this procedure ensures compliance, it could cause delays when adding new teachers and growing the instructor network for the app. To provide a clear picture of the system's limitations, it is critical to identify and address these deficiencies.



https://centennialcollegeedu-my.sharepoint.com/:u:/g/personal/jhuan145_my_centennialcollege_ca/EY4Pg4T08kxLg2_WFovnx3QBGQ2Qwz5mZ5ew3zRpoGFI8g?e=N6H3B0&action=edit&or=PrevEdit

2.3 User Role Modelling

2.3.1 Brainstorm and Group

During initial brainstorming, all user roles and groups were considered. Therefore, the initial set can be considered overcomplete.

Following the brainstorming of the overcomplete set of roles, there were 4 clusters containing a total of 14 roles. These were organized by similar characteristics in context of our application

The figure for the initial set and organization of the set is properly captioned and accompanied by a discussion. The caption provides a clear and concise description of the content of the figure, accurately representing what is depicted, with any errors and issues fixed throughout the workshop. Additionally, the discussion accompanying the figure provides an analysis and explanation of the key elements and relationships within the set.

The number of user roles in the consolidated set is minimized; many roles that were considered similar due to their characteristics were either merged together or removed altogether, reducing the number from 14 to 6. Clusters were reorganized, which included the removal of unused clusters.

The consolidated set of user roles is properly captioned such that it provides a explanation of the role and all roles are grouped in clusters with each cluster properly labeled to provide a general view of the contents

The descriptions for each user role are discussed throughout the workshop, especially during the creation of the consolidated set.

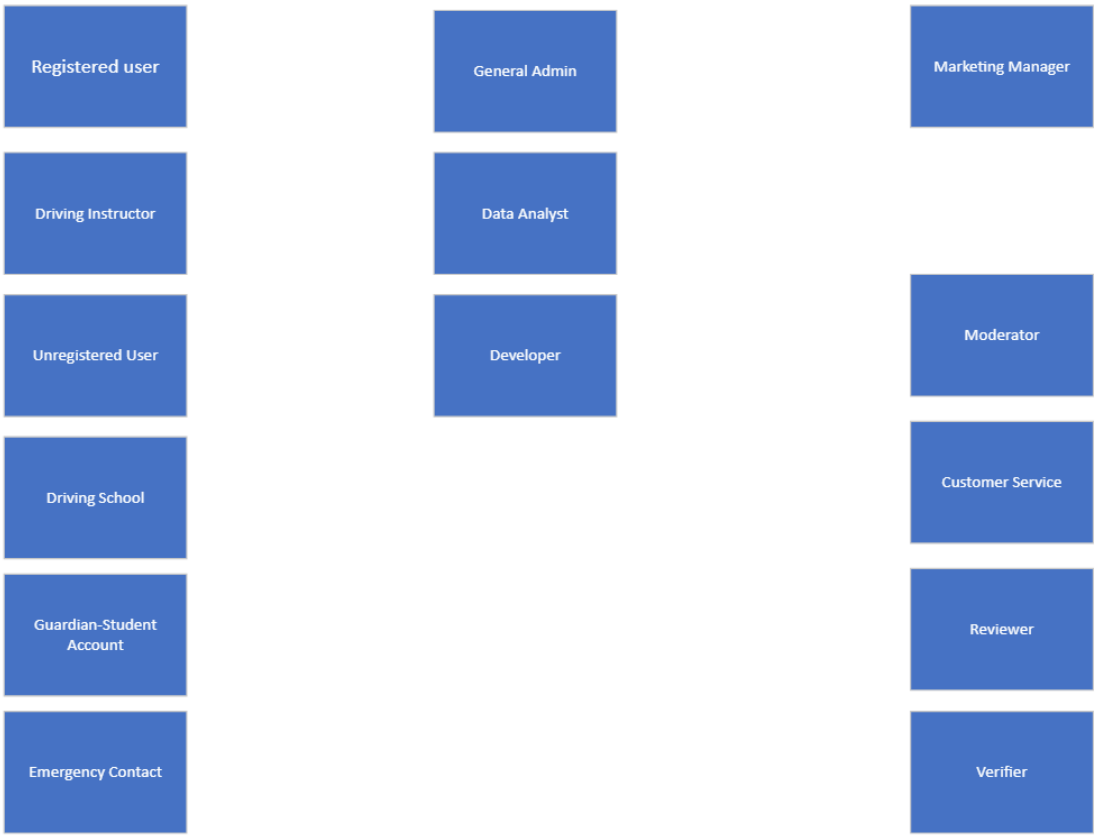


Figure 1: Organizing the user role cards on a table [1].

2.3.2 Consolidated User Roles

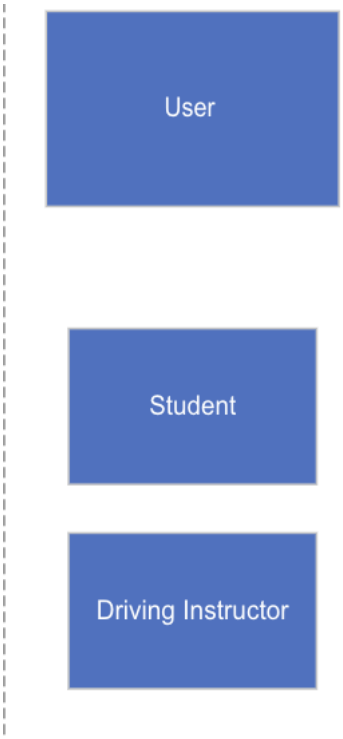


Figure 2: The consolidated role cards [1].

2.3.3 Description of User Roles and Persona

Role	**User**
Frequency	Somewhat Often
Expertise in domain	Dependent on the user
General Proficiency	Dependent on the user
Proficiency with Software	Dependent on the user
User's General goal	To book for and take driving lessons in order to gain appropriate knowledge for receiving their driving license

Role	**Driving Instructor**
Frequency	Very Often
Expertise in domain	Experienced

General Proficiency	Dependent on the user
Proficiency with Software	Experienced
User's General goal	To book students and give driving lessons/knowledge to help students receive driving license

2.3.4 Additional Documentation

Week 3 - User Roles video link:

https://www.youtube.com/watch?v=jwZA95RWuP0

2.4 Release 1.0

2.4.1 User Stories

During the workshop, the Driverooers team created low-fidelity prototypes for both user roles:

Consolidated Low Fidelity Diagrams

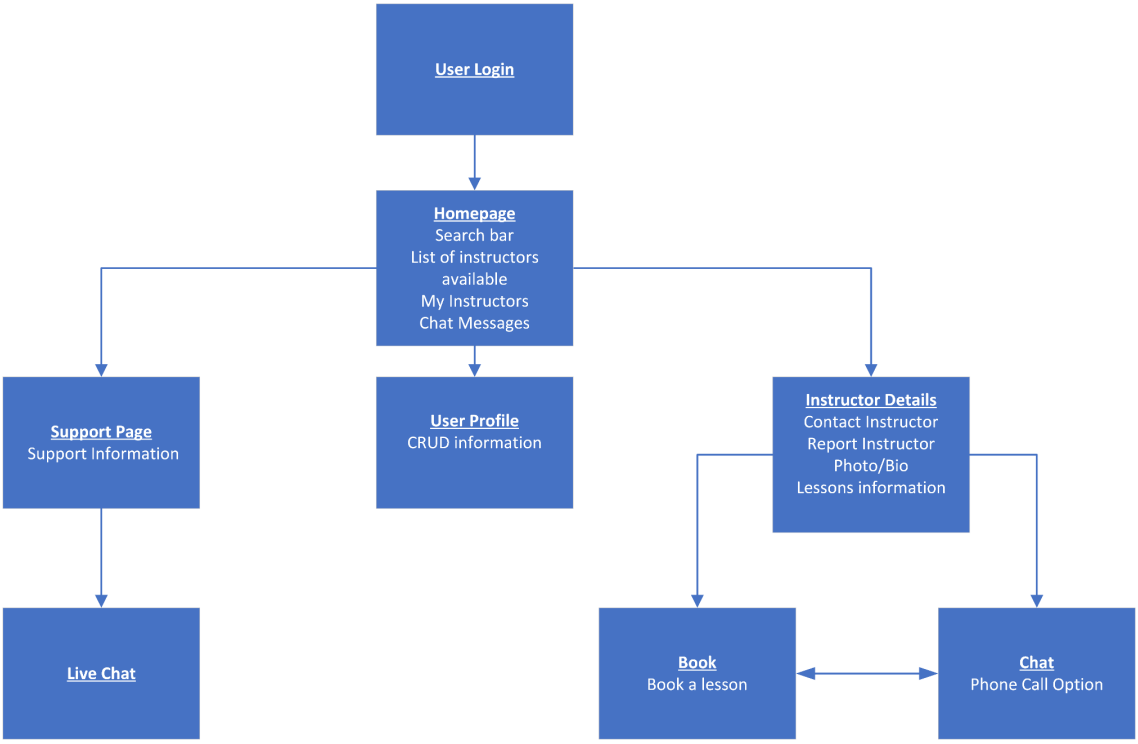


Figure 3 - Consolidated User Low-Fidelity Prototype Diagram

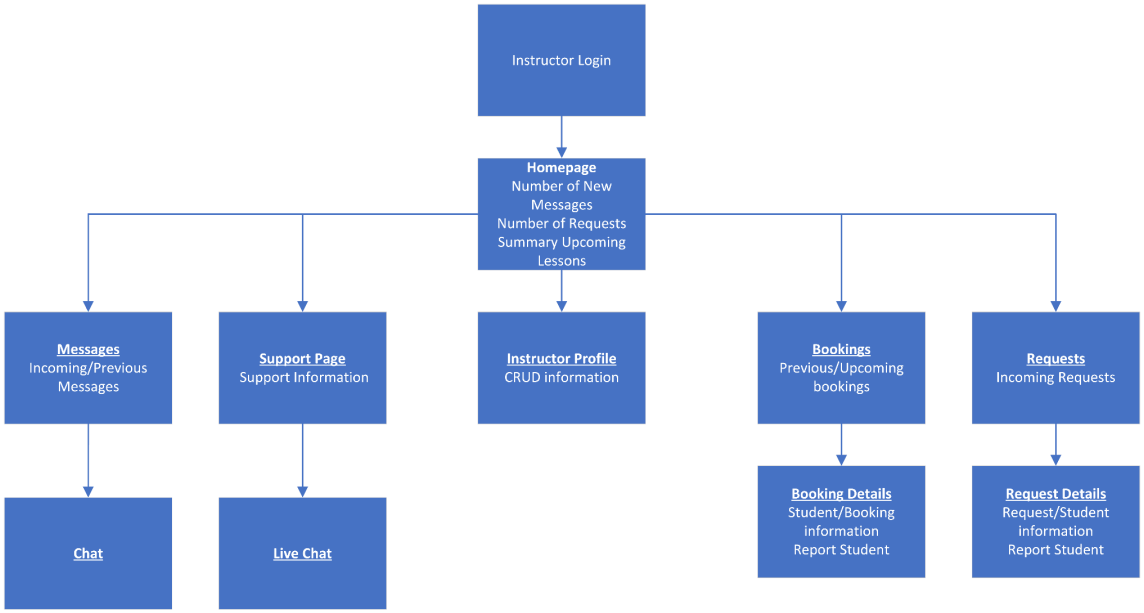


Figure 4 - Consolidated Instructor Low-Fidelity Prototype Diagram

The low-fidelity prototypes showcase a common progression for both user roles, requiring sign-in to access their respective homepages. Once signed in, users can update their personal profile information and utilize similar features, including messaging between different roles and customer service support.

Instructors, specifically, have access to their booking list and incoming requests for messages and bookings. They can also view detailed information about each booking and request, as well as engage in a chat box with users to discuss their specific needs.

Users, on the other hand, can browse available driving instructors' profiles and book lessons based on their availability. They can utilize the chat box to communicate with instructors and discuss their requirements during the lessons. Additionally, users can make online payments through the provided payment gateway to pay instructors in advance.

All user roles have access to a comprehensive homepage that offers an intuitive UI, enabling easy access to essential information such as messages, requests, and upcoming details.

For estimation purposes, one story point is equivalent to one hour of work for the development team.

A total of 20 User Story Cards were created and consolidated to cover the necessary functionality for the system.

As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.) **Estimated Time: 3 hrs**	**Acceptance Test:** test with fake user information **Expected Outcome:** result should show an error message saying invalid credentials, and tell you to sign up. **Acceptance Test:** test with valid user information **Expected Outcome:** redirect to home page with content for logged in users now available

As a user, I want to be able to view a list of my instructors, along with a search bar/sort for filtering, and a page consisting of previous chat rooms **Estimated Time: 36 hrs**	**Acceptance Test:** can find instructor via search bar **Expected Outcome:** result should consist of a list of the instructors with the same names and filter options **Acceptance Test:** test user can view list of old/ongoing conversations and send messages **Expected Outcome:** conversation can ensue between instructor and test user

As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught **Estimated Time: 12 hrs**	**Acceptance Test:** display details of a randomly selected instructor, and be able to engage in a chat with instructor **Expected Outcome:** result should feature photo, bio, and chat option **Acceptance Test:** test with reporting instructor **Expected Outcome:** report should be sent to admins

As a user, I want to be able to chat with the instructor **Estimated Time: 30 hrs**	**Acceptance Test:** test user can chat with the instructor and a phone call option should be working and available **Expected Outcome:** test user and instructor can hold conversation via phone or chat messages

As a user, I want to be able to book the instructors **Estimated Time: 35 hrs**	**Acceptance Test:** test user can book and pay for a lesson/package **Expected Outcome:** should add booking to their account and payment should go through

As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc. **Estimated Time: 8 hrs**	**Acceptance Test:** test with a test user, with fake information **Expected Outcome:** display correct information for the user retrieved from database **Acceptance Test:** update test user information **Expected Outcome:** information should be updated immediately.

	Acceptance Test: Delete test information **Expected Outcome:** information must be deleted both from app and database **Acceptance Test:** Delete test account and try logging in with same details **Expected Outcome:** Account should be deleted and user should be redirected to homepage

As a user, I want to be able to contact a support to help me with any issues **Estimated Time: 2 hrs**	**Acceptance Test:** test user should be able to see correct contact information and chat **Expected Outcome:** Display correct information and a button to initiate chat with customer service

As a user, I want to be able to hold a live-chat with customer service **Estimated Time: 30 hrs**	**Acceptance Test:** test user can initiate conversation with a real-human being **Expected Outcome:** conversation should ensue and issue should be resolved

As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.) **Estimated Time: 3 hrs**	**Acceptance Test:** test with fake user information **Expected Outcome:** result should show an error message saying invalid credentials, and tell you to sign up. **Acceptance Test:** test with valid user information **Expected Outcome:** redirect to home page with content for logged in users now available

As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests **Estimated Time: 14 hrs**	**Acceptance Test:** test with populating instructors requests with test requests **Expected Outcome:** instructor should be able to receive updated badge notification containing number of requests **Acceptance Test:** test with adding new lessons to schedule **Expected Outcome:** lessons should be added to the weekly overview **Acceptance Test:** instructor can view messages from users **Expected Outcome:** instructor can reinitiate conversation with users

As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc. **Estimated Time: 8 hrs**	**Acceptance Test:** test with a test instructor, and fake information **Expected Outcome:** display correct information for the instructor retrieved from database **Acceptance Test:** update test instructor information **Expected Outcome:** Information should be updated immediately. **Acceptance Test:** Delete test information **Expected Outcome:** information must be deleted both from app and database **Acceptance Test:** Delete test account and try logging in with same details **Expected Outcome:** Account should be deleted and user should be redirected to homepage

As an instructor, I want to be able to view previous and upcoming bookings **Estimated Time: 15 hrs**	**Acceptance Test:** test with a prior booking, with test information **Expected Outcome:** display time and date of the booking, including the name of student **Acceptance Test:** test with a new booking, with test information **Expected Outcome:** display time and date of the booking, including the name of student,

As an instructor, I want to be able to view details about any bookings or report the request/student **Estimated Time: 11 hrs**	**Acceptance Test:** test with a booking, with test information **Expected Outcome:** display time and date of the booking, including the name of student and any other necessary information **Acceptance Test:** test with reporting student **Expected Outcome:** report should be

As an instructor, I want to be able to view incoming requests for bookings **Estimated Time: 15 hrs**	**Acceptance Test:** test with a request for booking, with test information **Expected Outcome:** display overview, including time and date of the booking, and the name of student

As an instructor, I want to be able to view detailed information about a request or report the request/student **Estimated Time: 23 hrs**	**Acceptance Test:** test with a request for booking, with test information **Expected Outcome:** display detailed view including time and date of the booking, including the name of

	student, along with any other information **Acceptance Test:** test with reporting student **Expected Outcome:** report should be sent to the admin **Acceptance Test:** test with accepting/declining request **Expected Outcome:** request should be removed from the requests page and moved to booking

As an instructor, I want to be able to contact support to help me with any issues **Estimated Time: 2 hrs**	**Acceptance Test:** test user should be able to see correct contact information and chat **Expected Outcome:** Display correct information and a button to initiate chat with customer service

As an Instructor, I want to be able to hold a live-chat with customer service **Estimated Time: 30 hrs**	**Acceptance Test:** test user can initiate conversation with a real-human being **Expected Outcome:** conversation should ensue and issue should be resolved

As an Instructor, I want to be able to view any previous or incoming messages from students **Estimated Time: 8 hrs**	**Acceptance Test:**test with receiving new message **Expected Outcome:** instructor should receive message and go to chat page **Acceptance Test:**test if old/read messages are added to list of messages

	Expected Outcome:messages should be visible.

As an Instructor, I want to be able to chat with students **Estimated Time: 30hrs**	**Acceptance Test:**test with sending and receiving messages to a student **Expected Outcome:** student and instructor should be able to engage in conversation

Furthermore, the stories were prioritized using MoSCoW rules into tables consisting of must-haves and should-haves

Must-Haves	
Story	**Estimate**
As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)	3
As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.)	3
As a user, I want to be able to view a list of (my) instructors along with a search bar/sort for filtering, and a page consisting of previous chat rooms	36
As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught	12
As a user, I want to be able to book the instructors	35
As a user, I want to be able to view/update my information, such as	8

name, email, phone-number, payment information, etc.	
As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests	14
As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.	8
As an instructor, I want to be able to view previous and upcoming bookings	15
As an instructor, I want to be able to view details about any bookings or report the request/student	11
As an instructor, I want to be able to view incoming requests for bookings	15
As an instructor, I want to be able to view detailed information about a request or report the request/student	23
As an Instructor, I want to be able to view any previous or incoming messages from students	**8**

Table 1: MoSCoW - Must Haves

Should-Haves	
Story	**Estimate**
As an Instructor, I want to be able to chat with students	30
As a user, I want to be able to chat with the instructor	30
As a user, I want to be able to hold a live-chat with customer service	30

As a user, I want to be able to contact support to help me with any issues	2
As an Instructor, I want to be able to hold a live-chat with customer service	30
As an Instructor, I want to be able to chat with students	30

Table 2: MoSCoW - Should Haves

2.4.2 Additional Documentation

https://youtu.be/DMKk48kpI9o

2.4.3 Release Plan 1.0

The following are required for this section[1]:

1. Provide the product development roadmap.
2. Provide the iteration length and the release date.
3. The refine priorities of the Must- and Should-Have stories by organizing the stories into groups that have a high likelihood of being performed together.
4. The actual release plan.
5. Place the contents of your paper prototype in Appendix A (Design Document).

[1] See *The Release Plan* deliverable.

2.4.4 Iteration Plan (Release 1.0)

The following are required for this section:

1. Present each iteration plan with tables showing disaggregated tasks per story; a sample is shown in Table 3. See also the *Planning an Iteration* deliverable.
2. Discuss any discrepancies between the estimated and actual ideal time required to complete the tasks for the Table mentioned above.

Table 3: Disaggregated tasks per story [1].

Task	Who	Estimate	Actual
Code basic search screen	Susan	6	4
Code advanced search screen	Susan	8	9
Code results screen	Jay	6	8
Write and tune SQL to query the database for basic searches	Susan	4	3
Write and tune SQL to query the database for advanced searches	Susan	8	12
Document new functionality in help system and user's guide	Shannon	2	2

2.4.5 Additional Documentation

For this section, include 1 of 4 videos from your Iteration Planning meetings (recall that you have a total of 4 Iteration Planning meetings)[2]:

1. Showing how your team disaggregated stories into their constituent tasks.
2. How developers on your team volunteer and take responsibilities for tasks.

Provide the file name and URL to the video(s) in your shared folder or YouTube channel.

[2] Indicate which iteration the video corresponds to. If you decide to submit a video in Release 1.0, then you do not need to include an *Additional Documentation* section for Release 2.0.

2.4.7 Acceptance Tests for Release 1.0

The following are required for this section:

1. A table of stories and their associated acceptance tests for this Release as shown below in the sample in Table 5.
2. The link to your video demo for Release 1.0 stored either in a cloud drive, or your YouTube channel.

Table 4: Stories, acceptance tests, and contributors for Release 1.0 (Green=Passed; Red=Failed).

Full description of user story	**Acceptance test(s)**	**Name(s) of contributing Developer(s)**
As an User, I can … so that ….[3]	Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ...	Susan Smith, Jay Johnson
As an Administrator, I can … so that ….[4]	Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ...	Susan Smith, Jay Johnson, Shannon Shore, George Gavinson
As an User, I can … so that ….	Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ...	Jay Johnson, Shannon Shore, George Gavinson
As an User, I can … so that ….[5]	Test with inputs …. Expected outcome: ...	Shannon Shore
As a Guest, I can … so that ….	Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ... Test with inputs …. Expected outcome: ...	Susan Smith, Jay Johnson, Shannon Shore, George Gavinson, Abbey Appleby, Brian Bolt

<Insert url to video demo of Release 1.0 here>

[3] Green colour code indicates that all tests passed successfully as intended.
[4] Red colour code indicates that at least one test unintendedly failed.
[5] When all tests for a given story fails, this may suggest that implementation of the story has not even begun and indicates poor planning on the part of the team.

2.5 Release 2.0

Release 2.0 has essentially the same structure as Release 1.0.

2.5.1 User Stories

If your team wrote enough stories to cover up to or beyond Release 2.0 during your first story-writing workshop as described in the *User Stories* section 2.4.1, then your team will not need to hold a second formal workshop.

If a second workshop was held, submission for this section is the same as section 2.4.1.

2.5.2 Additional Documentation

Include this section in your Technical Report only if your team required a second formal story-writing workshop. If a second workshop was held, submission for this section is the same as section 2.4.2.

2.5.3 Release Plan 2.0

The requirements for this section are the same as section 2.4.3; update or add sections if required.

2.5.4 Iteration Plan (Release 2.0)

The requirements for this section are the same as section 2.4.4.

2.5.5 Additional Documentation

This section is required ONLY IF your team submitted materials for section 2.4.5.

2.5.7 Acceptance Tests for Release 2.0

The requirements for this section follow the same requirements as in section 2.4.7 except acceptance testing is for stories allocated for Release 2.0 and incomplete stories subsequently moved from Release 1.0.

3.0 CONCLUSIONS

A conclusion interprets the data found in the Body. It is reasoned judgment and not opinions. Consider the variables. Relate cause and effect. Analyze, evaluate, make comparisons and contrasts. Base the conclusion on fact.

4.0 RECOMMENDATIONS

Recommendations are not required for all studies. They suggest a course of action and would generally be provided when there are additional areas for study, or if the reason for the TR was to determine the best action going forward.

CREDITS, LICENSE, AND REFERENCES

Credits

Provide any credits here. The following are examples:

Author of the template graphic layout : Hao Lac

<haolac.at.centennial@gmail.com>

Author of the template explanation text : John Doe

<john.doe@centennialcollege.ca>

License

State the license granted with your system. For example:

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License".

References

[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

APPENDIX A (DESIGN DOCUMENT)

Traditional approaches to software development, in contrast to that of Agile approaches, place a great deal of emphasis on upfront design. The Agile approach to design is quick sessions that seek the simplest solution and then incrementally build on that solution. A quick design session can include the use of CRC cards that can ultimately lead to the generation of UML diagrams.

Using Agile approaches to software development does not mean you are limited to using only Agile techniques. If you feel that a technique (e.g., use case or interaction design scenario) is more suitable, or better conveys the features of your system to your users, then use it.

In this section, you are required to submit and discuss the following:

- A paper prototype of your application/system.
- Any design work your team has done in developing your system including CRC cards, UML diagrams, ERD diagrams, use cases, interaction design scenario, etc.

APPENDIX B (TEST PLAN)

1.0 Introduction

1.0.1 Goals

Summarize the testing goals for the project.

1.0.2 Assumptions

Any assumptions which may affect the understanding or execution of this plan should be recorded here.

1.0.3 Risks And Assets

Describe the elements (software or hardware) that are not part of your application but still may impact its correctness and must be checked.

Describe the elements that might positively influence testing on the project.

2.0 Scope

2.0.1 Features To Be Tested

Describe the features and functions that will be tested during the project. This should include functional and non-functional requirements.

2.0.2 Features Not To Be Tested

Describe the features that will not be tested and reason why.

3.0 Testing Procedures

Describe the testing procedures that the project will use. This includes the test lifecycle, types of testing, test objectives, and test criteria.

3.0.1 Test Objectives

Describe the objectives of the testing process.

3.0.2 Types Of Testing

Describe the types of testing that the project will use.

3.0.2.1 Unit Testing

Describe the strategy for unit testing of the individual subsystems. This includes an indication of the subsystems that will undergo unit tests or the criteria to be used to select subsystems for unit test. Test cases are NOT included here.

3.0.2.2 Integration Testing

Specify the integration testing strategy used. Describe the tests that will be performed in order to verify the interfaces between the subsystems of the software system. This section includes a discussion of the order of integration of subsystems. Test cases are NOT included here.

3.0.2.3 Acceptance Testing

Specify the strategy for testing the software once it has been deployed. This section includes a discussion of the order of acceptance by software function. Test cases are NOT included here.

3.0.2.4 Stress Testing

Identify the limits under which the program is expected to perform (memory constraints, disk space constraints, etc).

3.0.2.5 Performance Testing

Refer to the functional requirements that specify acceptable performance.

3.0.3 Testing Tools

Describe the tools that you will use for testing.

4.0 Schedule and Deliverables

Describe the test deliverables that will be created during the project lifecycle.

Include two tables, one for the schedule of tasks, another for the list of deliverables:

- Acceptance test
- Unit test
- System/Integration test
- Stress test
- Performance test
- Screen prototypes
- Defect reports and summaries
- Test logs and reports

Describe the reports that will be generated by the testing process.

Examples include:

Test Summary Report - A final report of the testing results from the project. Can

include items such as total number of test cases, number of test cases executed, % test cases passed, etc.

APPENDIX C (END-USER & ADMINISTRATOR MANUALS)

In this section, include a user manual for your system/application. The user manual should include the following items:

1. Instructions on how to install and configure your system/application, documenting all external software dependencies that need to be setup manually.
2. A user guide for the administrator (use screen shots of your system/application and briefly discuss each screen shot).
3. A user guide for the normal user (use screen shots of your system/application and briefly discuss each screen shot).

APPENDIX D (PROGRESS MONITORING)

Your team is required to report two items related to progress monitoring in this appendix. The first item is a table summarizing progress and changes during a release with supporting discussion; a sample is shown in Table 5. Notice in Table 5 that all iterations are shown per Release[6]. Also, see *Table 1* in the *Measuring and Monitoring Progress* deliverable.

Table 5: Progress and changes for all iterations [1].

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Story points at start of iteration	130	113	78	31
Completed during iteration	45	47	48	31
Changed estimates	10	4	–3	
Story points from new stories	18	8	4	
Story points at end of iteration	**113**	**78**	**31**	**0**

The second item is an iteration burndown chart (see Figure 9) reflecting the data from Table 5.

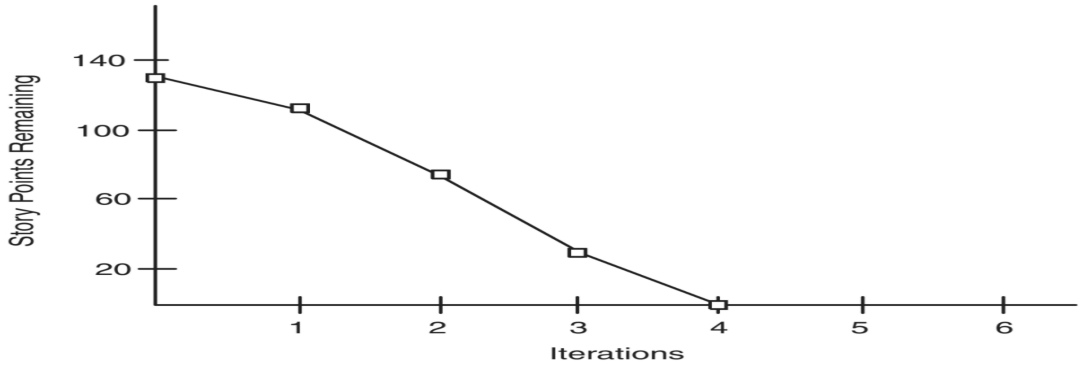


Figure 9: Iteration burndown chart for data from Table 5.

[6] For subsequent Releases, do NOT restart numbering the Iteration. For example, let us assume that we have another Release (i.e., Release 2.0), we would continue numbering our Iterations as *Iteration 5, Iteration 6,* and so on.